

TVORBA VÝUKOVÝCH SIMULÁTORŮ. CO JE ZA OPONOU DEVELOPMENT OF EDUCATIONAL SIMULATORS. WHAT IS BEHIND THE CURTAIN?

14

J. Kofránek

Laboratoř biokybernetiky, Ústav patologické fyziologie 1. LF UK, Praha

Abstrakt

Autor popisuje metodologii určenou pro vytváření e-learningového obsahu s využitím simulačních her. Je popsán celý návrhový proces, od sestavení formalizovaného popisu fyziologické reality, až vytváření interaktivního výukového softwaru. Při tom se využívají nejrůznější nástroje – od softwarových nástrojů pro numerickou simulaci jako Matlab/Simulink nebo Modelica, přes Adobe Flash pro tvorbu interaktivních animací, Control Web a Microsoft Visual Studio .NET pro vytváření uživatelského rozhraní, až po nástroje pro publikování na webu včetně Adobe Brezeze. Tvorba moderních výukových aplikací je náročný a komplikovaný projekt, vyžadující týmovou spolupráci řady profesí – pedagogů, lékařů, tvůrců simulačních modelů, výtvarníků a programátorů. Cílem je poskytnout studentům výukový software, který jim pomůže pochopit složitou dynamiku fyziologických systémů.

Klíčová slova

e-learning, fyziologie, počítačem podporovaná výuka, simulační modely, simulační hry

Abstract

The authors present a methodology of creating e-learning content with simulation games for the students of physiology and pathophysiology courses. The design process is explained starting from a formalized description of the physiological reality up to interactive educational software. Various software tools are used during the design, i.e. numerical simulation software Matlab/Simulink or Modelica, Adobe Flash for interactive animations, Control Web or Microsoft Visual Studio.NET for user interface design, and web publishing tools including the Adobe Breeze learning management system. Various professions are involved in complicated projects of modern educational applications – teachers, physicians, simulation/modeling experts, graphic designers and programmers. The aim is to provide students with software that helps them understand complex dynamics of physiological systems.

Keywords

computer aided learning, e-learning, medicine, physiology, simulation games, simulation models

Místo úvodu – pavučina fyziologických regulací

Před šestatřiceti lety vyšel v časopise časopise Annual of Biomedical Physiology článek [8], který se svou podobou na již první pohled naprosto vymyká navykly podobě fyziologických článků té doby. Byl uveden rozsáhlým schématem na vlepené příloze. Schéma plně čar a propojených prvků na první pohled vzdáleně připomínalo nákres nějakého elektrotechnického zařízení (obr. 1). Místo elektronek či jiných elektrotechnických součástek však zde byly zobrazeny propojené výpočetní bloky (násobičky, děličky, sumátory, integrátory, funkční bloky), které symbolizovaly matematické operace prováděné s fyziologickými veličinami (obr. 2). Svazky propojovacích vodičů mezi bloky na první pohled vyjadřovaly složité zpětnovazebné propojení fyziologických veličin. Bloky byly seskupeny do osmnácti skupin, které představovaly jednotlivé propojené fyziologické subsystémy. Centrálním byl subsystém reprezentující cirkulační dynamiku – s ním byly do jednoho celku zpětnovazebně provázány ostatní bloky: od ledvin, přes tkáňové tekutiny, elektrolyty, až po autonomní nervovou regulaci a hormonální řízení zahrnující ADH, angiotenzin a aldosteron (obr. 3).

Vlastní článek tímto tehdy naprosto novým způsobem pomocí matematického modelu popisoval fyziologické regulace cirkulačního systému a jeho širší fyziologické souvislosti a návaznosti na ostatní subsystémy organismu – ledviny, regulaci objemové a elektrolytové

rovnováhy aj. Místo vypisování soustavy matematických rovnic se v článku využívalo grafické znázornění matematických vztahů. Tato syntaxe umožnila graficky zobrazit souvislosti mezi jednotlivými fyziologickými veličinami ve formě propojených bloků reprezentujících matematické operace. Vlastní popis modelu byl pouze ve formě základního (ale přesto plně ilustrativního) obrázku. Komentáře a zdůvodnění formulací matematických vztahů byly velmi stručné: např. „bloky 266 až 270 počítají vliv buněčného PO₂, autonomní stimulace a bazální rychlosti spotřeby kyslíku tkáněmi na skutečnou rychlost spotřeby kyslíku v tkáních“. Od čtenáře to vyžadovalo nadměrnou velké soustředění (i jisté fyziologické a matematické znalosti) pro pochopení smyslu formalizovaných vztahů mezi fyziologickými veličinami.

O rok později, v roce 1973, vyšla monografie [9], kde byla řada použitých přístupů vysvětlena poněkud podrobněji. Guytonův model byl prvním rozsáhlým matematickým popisem fyziologických funkcí propojených subsystémů organismu a odstartoval oblast fyziologického výzkumu, která je dnes někdy popisována jako integrativní fyziologie. Z tohoto hlediska byl určitým mezníkem, který se snažil systémovým pohledem na fyziologické regulace zachytit dynamiku vztahů mezi regulací oběhu, ledvin, dýchání, objemu a iontového složení tělních tekutin pomocí matematického modelu.

Model neměl jen čistě teoretický význam – Guyton si záhy uvědomil i velký význam využití modelů jako svébytné výukové pomůcky.

Guyton a jeho žáci model nepřetržitě dále rozvíjeli. V roce 1982 Guytonův žák a spolupracovník Thomas Coleman vytvořil model "Human", určený především k výukovým účelům. Model umožnil simulovat řadu patologických stavů (kardiální a renální selhání, hemoragický šok aj.) i vliv některých terapeutických zásahů (transfúzi krve, umělou plicní ventilaci, dialýzu atd.) [5]. Moderní interaktivní webovou implementaci tohoto modelu v jazyce Java je možné najít na adrese <http://venus.skidmore.edu/human>.

Posledním výsledkem Guytonových žáků a následovníků je simulátor Quantitative Human Physiology [1]. Model je možné stáhnout z <http://physiology.umc.edu/themodelingworkshop/>.

Škola (simulační) hrou

Rozvoj osobních počítačů a internetu vnesl zcela nové možnosti pro praktické uplatnění simulačních modelů. Simulační model nemusí být jen nástrojem pro vědecký výzkum, může být i velmi užitečnou výukovou pomůckou.

Výukové programy se simulačními komponentami **nejsou jen multimediální náhradou klasických učebnic**. Jsou zcela **novou výukovou pomůckou**, kde nachází své moderní uplatnění staré krédo Jana Ámose Komenského "Schola Ludus" (škola hrou), které tento evropský pedagog razil již v 17. století. Spojení multimediálního prostředí se simulačními modely studentům umožňuje pomocí experimentů se simulačním modelem názorně prozkoumat vykládaný problém ve virtuální realitě a přináší tak zcela nové možnosti pro vysvětlování složitých problémů (obr. 4). Simulační hrou je možné bez rizika otestovat chování simulovaného objektu – např. zkusit přistávat virtuálním letadlem nebo, v případě lékařských simulátorů, léčit virtuálního pacienta, či otestovat chování jednotlivých fyziologických subsystémů. Simulační hry umožňují názorně vysvětlit komplexní vztahy ve fyziologických regulačních systémech a kauzální řetězce v patogenezi nejrůznějších onemocnění.

Z pedagogického hlediska je důležité, že modelovaný objekt můžeme rozdělit na jednotlivé subsystémy a testovat jejich chování odděleně i jako součást vyššího celku. Tak např. při studiu složitých fyziologických regulací můžeme dočasně odpojit vybrané regulační smyčky a umožnit studentům sledovat reakce těchto subsystémů na změny vstupních veličin (které jsou v reálném organismu ovšem samy regulovány). Tím dovolíme sledovat dynamiku chování jednotlivých subsystémů při postupných změnách pouze jediného vstupu, zatímco jiné vstupy jsou nastaveny na zvolenou konstantní hodnotu (tzv. princip "ceteris paribus").

Postupně pak můžeme jednotlivé dočasně rozpojené regulační vazby opět zapojovat a studovat jejich vliv na chování organismu při nejrůznějších patologických poruchách a reakcích na příslušnou terapii. Podle našich zkušeností právě tento přístup vede k lepšímu pochopení složitých dynamických jevů v patogenezi nejrůznějších onemocnění a porozumění patofyziologických principů příslušných léčebných zásahů.

Tak například v našem simulátoru Golem [14] můžeme postupně odpojovat a zapojovat jednotlivé regulační

smyčky ovlivňující homeostázu vnitřního prostředí a tím postupně testovat význam jednotlivých regulačních okruhů a ozřejmit si tím například souvislosti poruch iontové a acidobazické rovnováhy (viz obr. 5).

Vývoj efektivních výukových programů, kombinujících multimédia se simulačními hrami, je náročnou a komplikovanou prací, vyžadující týmovou spolupráci řady profesí – zkušených pedagogů vytvářejících základní scénář, tvůrců simulačních modelů, lékařů, výtvarníků a programátorů. Tuto interdisciplinární kolektivní tvorbu zefektivňuje využívání vhodných vývojových nástrojů, které umožňují komponentovou tvorbu, propojení simulačních programů a interaktivních multimédií podle daného scénáře do kompaktního celku.

Jedním z projektů, kde chceme využít nové možnosti multimédií a simulačních modelů pro výuku je počítačový **Atlas fyziologie a patofyziologie** [18], koncipovaný jako multimediální výuková pomůcka, která názornou cestou prostřednictvím Internetu s využitím simulačních modelů by měla pomoci vysvětlit funkci jednotlivých fyziologických systémů, příčiny a projevy jejich poruch – viz <http://physiome.cz/atlas>. Atlas je vytvářen jako společné dílo tvůrčího týmu odborníků různých profesí – lékařů, programátorů, systémových inženýrů a výtvarníků. **Projekt atlasu je otevřený** – jeho výsledky v českém jazyce chceme na internetu volně zpřístupnit všem zájemcům, a při jeho vývoji uvítáme spolupráci se všemi, kdo se budou chtít podílet na jeho postupném budování.

Ukažme si nyní, co vytváření takové interaktivní výukové pomůcky vyžaduje, neboli – "co je za oponou" tvorby výukových simulátorů.

Scénáristé výukových programů

Obdobně jako sebelepší učebnice zcela nenahradí učitele, tak ho nevytlačí ani seberafinovanější výukový program. A stejně tak jako klíčovým faktorem pro úspěch vysokoškolských učebnic je schopnost jejich autorů názorně a věcně správně vyložit složitou látku, tak pro zdar výukového programu je nejdůležitější jeho **kvalitní scénář**. Napsat dobrý učební text vysvětlující složité procesy však není jednoduché a veškeré pomůcky (multimédia, interaktivní animace, simulační modely apod.) – tedy vše co si lze představit pod donekonečna omílaným slovem e learning, jsou bez dobrého scénáře vlastního výukového programu či bez zkušeného učitele, který ví jak složitý simulační model pedagogicky využít, jen pouhou módní ozdobou.

V případě využití multimediálních animací je podkladem pro tvorbu scénáře nejen (hyper)textová část, ale, obdobně jako při produkci kresleného filmu, i jakýsi **"storyboard"** naznačující výtvarníkům jaké mají vytvářet animované obrázky.

Složitější to je ovšem při využití simulačních modelů. Tam nestačí jen namalovat návrhy obrázků. Je zapotřebí nejprve **vytvořit vlastní simulační model**. Obdobně, jako je teoretickým podkladem letového simulátoru více méně realistický matematický model letadla, je v pozadí lékařského simulátoru matematický model lidského organismu (resp. nějakého jeho subsystému).

Formalizace fyziologických vztahů

Při studiu dynamiky regulačních vztahů a pochopení vzájemných dynamických souvislostí fyziologických regulačních vztahů často jen slovní popis a "selský

rozum" nestačí. Pro kvantifikované vyjádření dynamických vztahů je zapotřebí tyto vztahy **formalizovat**, tj. vyjádřit je pomocí rovnic matematického modelu.

Formalizace, tj. převedení čisté verbálního popisu příslušné sítě vztahů na popis ve formalizovaném jazyce matematiky, umožňuje zkoumat chování formálně popsaného systému pomocí formálních pravidel – např. řešením rovnic matematického modelu.

Obdobně jako fyzika popisuje přírodní zákonitosti pomocí matematických vzorců a soustav rovnic (a popis chování fyzikálního světa je výsledkem řešení těchto rovnic), tak i systémový fyziolog se snaží vyjádřit fyziologické regulace pomocí matematických vztahů jako matematický model fyziologické reality. To obvykle vede na popis prostřednictvím soustavy diferenciálních rovnic (jejichž "ruční" řešení bývá obtížné a často až prakticky nemožné). Pokud ale tyto rovnice převedeme do formy programu pro počítač, můžeme pak přenechat "dřinu strojům" – a na počítači sledovat průběhy hodnot jednotlivých proměnných, které zobrazují chování modelovaného systému jako výsledek řešení rovnic matematického modelu. Na počítači můžeme provádět s modelem nejrůznější experimenty. Průběhy hodnot těchto proměnných v čase nahrazují – tj. simulují chování skutečného fyziologického organismu. V tom je právě podstata využití tzv. **simulačních modelů** (tj. matematických modelů, implementovaných na počítači).

Simulační model, pochopitelně, nenahradí biologický experiment (jak tvrdí někteří fanatičtí bojovníci proti pokusům na zvířatech). Simulační model je ale velice efektivním nástrojem na dedukci a ověřování hypotéz – umožňuje sledovat chování složitého dynamického systému v čase v závislosti na nejrůznějších vstupech a porovnávat průběhy hodnot jednotlivých proměnných modelu s hodnotami získanými pozorováním a měřením biologického originálu.

Kritériem pro odmítnutí nebo neodmítnutí hypotézy je vždy porovnání chování modelu s empiricky a experimentálně získaným chováním biologického originálu. Pokud dosáhneme (na zvolené úrovni přesnosti) dostatečné shody chování modelu s chováním biologického originálu (hovoříme o tzv. verifikaci modelu), pak můžeme (s předem jasným omezením přesnosti) využít model k predikci – např. v klinice pro propočítávání kinetiky podaných léků nebo k propočítávání dekompresních zastávek při sportovním potápění atd.

Je to obdobně jako ve fyzice – pokud chování fyzikálního modelu dobře predikuje (na dané úrovni přesnosti) chování fyzikálních experimentů, můžeme dané vztahy ve fyzikálních vzorcích využít pro nějaké praktické účely – např. pro počítání statiky budov a další technické aplikace.

Fyzika využívá formalizace dávno – již od dob Newtona – a uplatnění formalizovaného vyjádření fyzikálních zákonů je podstatou moderních technických aplikací. Uplatnění formalizace v biologických a lékařských vědách však ani zdaleka není běžné. Oproti technickým vědám, fyzice či chemii má zde biologie a medicína určitý handicap. Jestliže formalizace ve fyzice začala již někdy v sedmáctém století, proces formalizace lékařských a biologických věd je z důvodů složitosti a komplexnosti biologických systémů relativně opožděn a postupně přichází až s kybernetikou a výpočetní technikou.

Ve fyziologii se s formalizovanými popisy setkáváme již od konce šedesátých let dvacátého století (od průkop-

nických prací Grodinse a spol., [7] popisujících respirační). Předělem byl počátek sedmdesátých let, kdy rozvoj výpočetní techniky a programovacích jazyků ve větší míře umožnil praktické vytváření a testování simulačních modelů fyziologických systémů. Určitým mezníkem byl již v úvodu zmíněný popis cirkulace Guytonem a spol. z roku 1972 [8], který byl prvním rozsáhlejším modelem, snažícím se zachytit širší fyziologické souvislosti oběhového systému, dýchání a ledvin. Koncem sedmdesátých let se v odborné literatuře postupně objevují rozsáhlé simulační modely, snažící se integrativně zachytit spletité vztahy mezi regulací ledvin, dýchání, oběhu, iontového složení, acidobazické rovnováhy a dynamiky tělních tekutin pomocí soustav nelineárních diferenciálních rovnic [2, 4, 13].

Bouřlivý rozvoj modelování biologických systémů však přichází zejména v posledních letech v souvislosti s novými možnostmi, které přináší technický pokrok v oblasti výpočetní techniky a Internetu. Stoupá počet prací využívajících počítačové modely **pro vyhodnocování a interpretaci výsledků experimentálních dat** zejména při studiu nervové tkáně, respirace, oběhu a ledvin. Byl odstartován velký rozvoj v oblasti formalizace fyziologických systémů.

Tak jako teoretická fyzika se snaží interpretovat výsledky experimentálního výzkumu ve fyzikálních vědách, tak i nový fyziologický směr základního výzkumu, nazývaný někdy "**integrativní fyziologie**", se snaží o **formalizovaný popis vzájemného propojení fyziologických regulací**. Metodickým nástrojem jsou zde počítačové modely. Aktivity v této oblasti se snaží koncentrovat projekt PHYSIOME [3, 11, 12] pod záštitou International Union of Physiological Sciences (<http://www.physiome.org>). V rámci Evropské Unie je současně době vyhlášen celoevropský projekt EUROPHYSIOME (<http://www.europphysiome.org/>). Jedním z cílů tohoto projektu je využití simulačních modelů pro výuku.

Počítačové modely z oblasti biologie a medicíny nyní stále častěji nacházejí přímé praktické uplatnění v sofistikovaných výukových programech a lékařských simulátorech, jejichž význam s rozšiřováním počítačů a Internetu dále poroste.

Dva typy problémů při tvorbě výukových simulátorů

Při vytváření simulátorů a výukových simulačních her je nutno řešit dva typy problémů:

1. Tvorba simulačního modelu – vlastní teoretická výzkumná práce, jejímž podkladem je formalizace fyziologických vztahů vyjádřená matematickým modelem. Výsledkem by měl být verifikovaný simulační model, který na zvolené úrovni přesnosti dostatečně věrně odráží chování modelované reality.
2. Tvorba vlastního multimediálního simulátoru, resp. tvorba výukového programu využívajícího simulační hry – je praktická aplikace teoretických výsledků, která navazuje na výsledky řešení výzkumu. Podkladem simulátoru jsou vytvořené (a verifikované) matematické modely. Zde jde o náročnou vývojovou práci, vyžadující skloubit nápady a zkušenosti pedagogů, vytvářejících scénář výukového programu, kreativitu výtvarníků, vytvářejících interaktivní multimediální komponenty a úsilí programátorů, kteří "sešijí" výsledné dílo do konečné podoby.

Každý z těchto problémů má svou specifiku a vyžaduje proto použít zcela odlišné vývojové nástroje.

Zatímco vytvoření vlastního simulátoru je spíše vývojářskou a programátorskou prací, formalizace fyziologických vztahů a vytváření simulačního modelu není vývojářský, ale (poměrně náročný) výzkumný problém, jehož efektivní řešení vyžaduje použití adekvátní nástroje pro podporu tvorby simulačních modelů.

Tvorba simulačního modelu i tvorba simulátoru spolu úzce navzájem souvisejí (viz obr. 6) – předpokladem pro tvorbu výukového simulátoru je dostatečně dobře verifikovaný model, na druhé straně, využití simulátoru ve výuce přináší nové požadavky na vytvoření nových či modifikaci stávajících simulačních modelů.

A protože pro tvorbu simulačních modelů a pro vytváření vlastního simulátoru používáme odlišné vývojové nástroje, musíme pak zajistit dostatečně flexibilní přenos výsledků z jednoho vývojového prostředí do druhého. Tak např. modifikujeme-li simulační model v některém nástroji pro tvorbu simulačních modelů, je výhodné zajistit, aby se změny v modelu bez větších potíží mohly rychle promítnout do aktualizace těchto změn ve vlastním simulátoru. K usnadnění tohoto přenosu je vhodné si vytvořit i vlastní softwarové pomůcky.

V naší laboratoři jsme při vytváření a verifikaci simulačních modelů sáhli po profesionálních nástrojích firmy Mathworks (Matlab, Simulink a příslušné aplikační knihovny) a v poslední době i po velmi nadějném simulačním jazyku Modelica, pro něž dotváříme i některé vlastní komponenty vývojového prostředí.

Při vytváření vlastního simulátoru využíváme standardní vývojová prostředí pro tvorbu a vývoj interaktivní grafiky (např. Adobe Flash) a nástroje pro tvorbu softwarových a webových komponent (např. Visual Studio .NET, prostředí pro vývoj v jazyce Java jako třeba NetBeans, prostředí pro návrh a programování interaktivní grafiky v jazyce ActionScript jako je např. Adobe Flex aj.). V minulosti jsme výukové simulátory vyvíjeli také pomocí nástroje pro vizualizaci průmyslových aplikací Control Web.

Pro propojení obou skupin vývojových nástrojů jsme vyvinuli vlastní softwarové nástroje pro automatizaci přenosu simulačních modelů, vytvořených v prostředí Matlab/Simulink do vývojových prostředí Microsoft Visual Studio .NET a do Control Web [17, 22]. V současné době vyvíjíme softwarové nástroje pro zefektivnění tvorby simulátorů založených na simulačních modelech vyvíjených v prostředí simulačního jazyka Modelica. Tyto programy jsou k dispozici všem zájemcům na vyžádání (e-mailem na adresu autora).

Teoretický základ simulačních her – simulační modely

V sedmdesátých a osmdesátých letech minulého století autoři často volili pro grafický popis modelů fyziologických systémů grafickou syntaxi navrženou Guytonem – tak např. Amosov a spol. v roce 1977 [2] publikovali monografii v níž stejnou syntaxi popsali propojený model cirkulace, respirace, ledvin, elektrolytové rovnováhy a termoregulace. Obdobně v roce 1979 tuto syntaxi využil Ikeda a spol. [13] ve svém modelu vnitřního prostředí.

Později, s rozvojem výpočetní techniky a softwarových nástrojů, se pro grafické zobrazení fyziologických re-

gulačních vztahů začaly využívat nástroje simulačních vývojových prostředí.

Jedním z nich je např. vývojové prostředí Matlab/Simulink od firmy Mathworks, které umožňuje postupně sestavovat simulační model z jednotlivých komponent – jakýchsi softwarových simulačních součástek, které se pomocí počítačové myši se sebou propojují do simulačních sítí. Obdobně jako u výše zmíněného Guytonova modelu i tyto sítě svými vodiči a součástkami na první pohled vzdáleně připomínají elektronické obvody. Rozdíl je ovšem v tom, že místo elektrických signálů v propojovaných vodičích proudí informace. V jednotlivých uzlech (sumátorech, integrátorech, násobičkách, děličkách atd.) pak dochází ke zpracování přiváděných informací a k transformaci do výstupní informace, která je rozepisována výstupními vodiči do dalších prvků sítě.

V Simulinku tak můžeme názorně graficky vyjádřit i velmi složité matematické vztahy. Vytvořená síť ovšem není jen přehledný obrázek. Je to zároveň i funkční simulační model. K jednotlivým vodičům této sítě můžeme myši připojovat virtuální displeje či osciloskopy. Po spuštění modelu pak můžeme jejich prostřednictvím sledovat číselnou či grafickou podobu hodnot veličin, které proudí v jednotlivých vodičích.

Jako příklad můžeme uvést námi vytvořenou implementaci klasického Guytonova modelu cirkulace [19]. Vnější vzhled jsme se snažili zachovat zcela stejný, jako v původním grafickém schématu – rozložení, rozmístění vodičů, názvy veličin i čísla bloků jsou stejné. Simulinkové počítací prvky jsou velmi podobné prvkům, které pro formalizované vyjádření fyziologických vztahů použil Guyton (obr. 7). Rozdíl je jen v grafickém tvaru jednotlivých prvků – tak např. násobička a dělička je v Simulinku realizována čtverečkem a nikoli "prasátkem" jako v Guytonově notaci. Blok integrátoru zobrazený v Simulinku nemá na sobě grafický znak integrálu, ale výraz "1/s" (což souvisí s notací tzv. Laplaceovy transformace). V simulinkovém modelu jsme využili i přepínače, kterými můžeme odpojovat nebo zapojovat jednotlivé subsystémy a regulační smyčky i za běhu modelu.

Simulační vizualizace starého schématu ale zdaleka nebyla úplně snadná – v originálním obrázkovém schématu modelu jsou totiž chyby. V nakresleném obrázku to nevidí, pokusíme-li se ho ale oživit v Simulinku, pak se chyba projeví v lepším případě neadekvátním chováním modelu, v horším se model znestabilní, hodnoty proměnných modelu začnou divoce kmitat a model zkolabuje jako celek. Chyby byly drobné – přehozená znaménka, dělička místo násobičky, prohozené propojení mezi bloky, chybějící desetinná tečka u konstanty atd. – ale stačily na to, aby model nefungoval. Při znalosti fyziologie a systémové analýzy se na chyby, při troše námahy, dalo přijít (obr. 8).

Je zajímavé, že tento složitý obrázek byl mnohokrát přetiskován do různých publikací a nikdo si nedal práci tyto chyby odstranit. Ostatně, v době, kdy obrázkové schéma vznikalo ještě neexistovaly kreslicí programy – obrázek vznikl jako složitý výkres – a ruční překreslování složitějšího výkresu ani zdaleka není snadné. Možné je i to, že autoři opravovat chyby ani příliš nechtěli – kdo si dal práci s analýzou modelu, snadno obrazové "překlepy" odhalil, kdo by chtěl jen tupě opisovat, měl smůlu. Konec konců, ve své době autoři rozesílali i zdrojové texty

programů svého modelu v programovacím jazyce Fortran – takže pokud někdo chtěl pouze testovat chování modelu, nemusel nic programovat (maximálně pouze rutinně převést program z Fortranu do jiného programovacího jazyka).

Námi vytvořená Simulinková realizace (opraveného) Guytonova modelu (obr. 9–11) je zájemcům k dispozici ke stažení na adrese www.physiome.cz/guyton/. Na této adrese je i naše Simulinková realizace mnohem složitější verze modelu Guytona a spol. z pozdějších let. Zároveň je zde i velmi podrobný popis všech použitých matematických vztahů se zdůvodněním. K dispozici je i výpis zdrojového textu původního modelu z roku 1986 ve Fortranu.

Od simulační pavučiny k simulačním čipům

Spleťtá pavučina počítačích bloků Guytonova modelu, implementovaná v prostředí Simulinku působí sice svou složitostí na první pohled impozantně, pro vlastní práci s modelem je však dosti nepřehledná.

Při výstavbě našich vlastních modelů se nesnažíme stavět mraveniště bloků prošpikovaných svazky informačních vodičů, v nichž se uživatel (a nezřídka i sám tvůrce modelu) velmi snadno ztrácí. Snažíme se proto, pokud to jde, model dekomponovat do přehledných částí. Využíváme toho, že jednotlivé části simulační sítě se dají hierarchicky seskupovat do subsystémů s uživatelsky definovanými vstupy a výstupy, v masce subsystému se dají vstupy a výstupy stručně popsat. Na kliknutí je možno k subsystému zpřístupnit i rozsáhlou dokumentaci, kde je možno bez omezení podrobněji popsat i teoretické pozadí vytvořeného subsystému.

Takto pojaté subsystémy na první pohled připomínají elektronické integrované obvody, neboť na jednotlivých "pinech" těchto "simulačních čipů" jsou příslušné vstupy a výstupy, které je možno propojovat do sítě s dalšími bloky a subsystémy (obr. 12). Struktura simulačního modelu vyjádřeného jako síť propojených simulačních čipů je pak mnohem přehlednější a srozumitelnější (obr. 13, 14).

Důležité je, že nakreslené struktury jsou v prostředí programu Simulink "živé" – pomocí počítačové myši lze na jednotlivé vstupy jednotlivých simulačních čipů přivést konstantní nebo proměnné hodnoty vstupních veličin a na výstupech pak snímat v numerickém či grafickém vyjádření jednotlivé hodnoty výstupních proměnných.

Vnitřní struktura "čipu" může být při jeho použití zcela skryta, uživatel musí pouze vědět, jakou vstupní či výstupní veličinu příslušný "pin" reprezentuje. Pomocí myši může uživatel přenést čip z příslušné knihovny do vytvářeného modelu, kde příslušné vstupy a výstupy propojí. Chování "čipu" lze také **snadno testovat**, stačí k jeho vstupům přivést příslušné definované průběhy hodnot vstupních veličin a na jeho výstupy připojit příslušné virtuální osciloskopy či displeje.

Každý čip může "uvnitř sebe" obsahovat další propojené "simulační čipy". Obsáhlý simulační čip tak může mít poměrně složitou hierarchii (obr. 15–16).

Simulační čipy je možno ukládat do knihoven a opětovně používat v různých modelech (obr. 17).

Tvorba simulačních modelů v biomedicínských vědách je nezřídka týmová práce. Na jedné straně stojí **systémový analytik** – expert na formalizaci a tvorbu simu-

lačních modelů (teoretický fyziolog, vytvářející formalizovaný popis fyziologického systému a testující jeho chování pomocí simulačního modelu). Na druhé straně často stojí klasický **experimentální fyziolog či klinik**, pro kterého je popis fyziologického systému pomocí diferenciálních rovnic španělskou vesnicí, ale který dokáže snadno rozpoznat, nakolik odpovídá chování počítačového simulačního modelu biologické realitě.

Podle našich zkušeností může problém dorozumění mezi těmito dvěma skupinami specialistů zásadně usnadnit důsledné využívání simulačních čipů při výstavbě simulačního modelu. Znamená to v prvé řadě věnovat důslednou pozornost dokumentaci. **Simulační čipy** samy o sobě mohou být i **aktuální elektronickou dokumentací** k vytvářeným modelům (viz obr. 17). Nejenže v čelní masce simulačního čipu je stručný popis všech vstupů a výstupů a "vnitřek" simulačního čipu graficky reprezentuje síť použitých vztahů, na kliknutí myši lze u každého softwarového simulačního čipu také otevřít okno nápovědy s dalším podrobnějším popisem. Simulační čipy tedy umožňují pečlivě vést aktuální dokumentaci v elektronické podobě ke každému použitému subsystému při zachování veškeré funkčnosti. Aby mohl jednotlivé komponenty využívat i ten, kdo danou část modelu nevytváří, je podrobná a zároveň přehledná dokumentace nezbytná a čas strávený nad vypisováním spousty informací do masek jednotlivých subsystémů reprezentujících simulační čipy se pak zaručeně vyplatí [Kofránek a spol, 2002].

Výhodou za trochu té dřiny je porozumění – experimentální fyziolog nemusí rozumět vnitřnímu uspořádání simulačního čipu, porozumí ale tomu, jaké chování má od fyziologického subsystému, který čip reprezentuje, očekávat. Fyziolog je nadto schopen porozumět i struktuře složené z propojených simulačních čipů – ze struktury modelu přímo vidí, které veličiny spolu vzájemně souvisejí (a ze znalosti fyziologie i dovede odhadnout, co a na jaké úrovni bylo zanedbáno).

Od rovnic modelu k technologickému know how

V sedmdesátých a osmdesátých letech dvacátého století bylo časté, že autoři počítačových modelů na vyzádnání posílali zdrojové texty počítačových programů, kterými byly tyto modely realizovány.

Dnes se už zdrojové programy rozsáhlých simulačních programů získávají mnohem obtížněji. Rozvoj výpočetní techniky totiž umožnil praktické uplatnění simulačních modelů například v lékařských počítačových trenažérech a výukových programech využívajících simulační hry. Biomedicínské modely mají své místo i v technologických zařízeních – od kapesních počítačů pro sportovní potápěče až po řídicí jednotky v sofistikovaných lékařských přístrojích.

S možností komerčního využití souvisí i určitá změna náhledu na **formalizovaný popis fyziologické reality**, který je podkladem výukových simulačních modelů. Podrobný formalizovaný popis (ve formě rovnic, nebo, ještě lépe, algoritmů zapsaných ve formě zdrojového textu v nějakém programovacím jazyce) se proto z **čistě vědeckého tématu často stává** (před potenciální konkurencí utajovaným) **technologickým know how**. Jestliže ještě koncem osmdesátých let bylo běžné získat na písemnou žádost přímo výpis příslušného fortranového programu, nyní to již většinou není možné.

V naší laboratoři biokybernetiky se zabýváme tvorbou simulačních modelů v rámci výzkumu dotovaného z veřejných zdrojů i díky spolupráci s komerčním sektorem. Struktura námi vytvářených modelů, dotovaných z veřejných prostředků je veřejná. V rámci výzkumných grantů a rozvojových projektů jsme mimo jiné vytvořili knihovnu fyziologických modelů v prostředí Simulink-Matlab, ve formě "simulačních čipů". Tato knihovna byla vytvořena jako Open-Source produkt, byla periodicky aktualizována a je k dispozici zájemcům na našich internetových stránkách (www.physiome.cz/simchips).

Tvorba simulátorů pro simulační hry

Vývojové nástroje pro tvorbu simulačních modelů jsou určeny pro specialisty. Pro běžného uživatele, který si chce se simulačním modelem jen "pohrát", se příliš nehodí. I když v prostředí těchto nástrojů je možné naprogramovat poměrně příjemné uživatelské rozhraní k ovládní vytvořeného modelu, pro účely uplatnění simulačního modelu ve výuce medicíny je toto rozhraní až příliš komplikované a navíc vyžaduje zakoupení dalších (poměrně drahých) licencí.

Student medicíny a lékař vyžaduje uživatelské rozhraní simulátoru připomínající spíše obrázky a schémata obdobná, jako v knižních publikacích typu atlasu fyziologie či atlasu patologické fyziologie.

Proto je nutné výukový simulátor včetně jeho multimediálního uživatelského rozhraní naprogramovat zvlášť. Možnosti uživatelského ovládní simulátoru jsou pak pro cílovou skupinu uživatelů podstatně přirozenější. Nezastupitelnou komponentou výukového simulátoru je i část programu, která realizuje simulační model. Známe-li strukturu simulačního modelu, (vytvořeného v některém z vývojových nástrojů pro tvorbu simulačních modelů – např. Matlab/Simulink) pak už zbývá "hlavoruční dřinou" přetvořit strukturu modelu do podoby počítačového programu ve zvoleném programovacím jazyce (např. v C#).

Dále je zapotřebí pro vytvoření uživatelského rozhraní nakreslit interaktivní multimediální komponenty. Tyto komponenty je pak nutno propojit se simulačním modelem na pozadí simulátoru.

Pro vývoj simulátorů ve výukových programech jsme se proto museli poohlédnout po jiném vývojovém prostředí než je Matlab a Simulink. Po určitém váhání jsme pro tvorbu vlastních simulátorů nejprve sáhli po **nástrojích využívaných při tvorbě průmyslových aplikací** (měřících ústředn a velinů) [16]. Vedly nás k tomu především dva důvody:

1. Se simulačním modelem chceme v simulátoru fyziologických funkcí zacházet obdobně, jako se v průmyslu z velinu řídí složité technologické zařízení: chceme číst (a v nejrůznější grafické či číselné podobě zobrazovat) množství nejrůznějších měřených dat (jako v průmyslové měřící ústředně) a zároveň chceme jednoduchým způsobem (stiskem tlačítek, otáčením knoflíků, popotahováním táhel apod.) simulační model ovládat (obdobně jako se z velinu řídí nějaká technologie).
2. Druhým důvodem, proč jsme sáhli po softwarovém nástroji z průmyslu, je spolehlivost. Požadavky spolehlivosti, kladené na nástroje, jejichž pomocí se vyvíjejí průmyslové řídicí aplikace, jsou obvykle řádově vyšší než u obecných programovacích nástrojů.

Nástrojů pro design průmyslových aplikací je na světovém trhu nemálo. Jejich ceny jsou ovšem, na rozdíl od obecných softwarových nástrojů, zpravidla velmi vysoké.

Paradoxem je, že velmi dobrý nástroj lze získat velmi levně, a to přímo od domácího výrobce. Zlínská akciová společnost "**Moravské přístroje**" již léta vyvíjí systém pro tvorbu průmyslových aplikací s názvem "**Control Web**". Jde přitom o kvalitní vývojový systém: na letáčku jejich skandinávského distributora stojí "Svensk kvalitet (till Tjeckiska priser)" – "Švédská kvalita (za české ceny)". Švédové jsou velmi domýšliví na kvalitu svých výrobků a pokud o vývojovém nástroji **Control Web** z Valaška veřejně prohlašují, že dosahuje "švédské kvality", je to velké ocenění skupiny tvůrců, kteří na tomto produktu usilovně pracují již od počátku devadesátých let.

Control Web je především určen **pro vývoj průmyslových vizualizačních a řídicích aplikací na platformě WIN32** – sběr, ukládání a vyhodnocování dat, tvorba rozhraní člověk-stroj aj. (viz <http://www.mii.cz>). Základními stavebními kameny uživatelské aplikace vytvářené v prostředí Control Web jsou **virtuální přístroje** (komunikující mezi sebou pomocí proměnných a zpráv). Měřené hodnoty z vnějšího světa jsou v průmyslových aplikacích virtuálním přístrojem zprostředkovány přes **vstupní kanály, řídicí signály** mohou virtuální přístroje posílat do okolí pomocí **výstupních kanálů**.

Pro vývoj uživatelského rozhraní poskytuje systém Control Web velmi výkonné prostředky. Tak např. z palety virtuálních přístrojů je možno snadno tažením myši vytáhnout potřebný přístroj a umístit ho na příslušný panel a v interaktivním dialogu mu nastavit hodnoty jeho příslušných atributů, nadefinovat jeho lokální proměnné, či individuální procedury (metody objektu) apod.

Abychom mohli využít vývojářské pohodlí systému Control Web, museli jsme použít následující, vcelku jednoduchý, trik. V průmyslových aplikacích Control Web komunikuje (přes příslušné softwarové kanály) přes ovladač příslušné měřicí/řídicí karty s průmyslovým technologickým zařízením.

Je ovšem možné napsat **speciální ovladač**, jehož interní součástí je simulační model. Tento ovladač je schopen komunikovat (přes softwarové kanály) s objekty systému Control Web, ale na rozdíl od ovladačů ke skutečným měřicím a řídicím kartám nekomunikuje s jejich hardwarem, ale **komunikuje se simulačním modelem**. Pokud se ovladač napíše dobře, je systém Control Web "ošálen": vstupní kanály (k zobrazovacím prvkům na monitoru) považuje za skutečné měřené signály někde z technologického okolí počítače, zatímco ve skutečnosti to jsou výstupní proměnné simulačního modelu. A také v opačném směru je Control Web přesvědčen, že výstupní kanály, které odcházejí od řídicích prvků systému Control Web, nastavují přes příslušný ovladač nějaké aktivní prvky průmyslového zařízení, ale ony namísto toho mění vstupy simulačního modelu.

Abychom usnadnili vývoj ovladačů jakési "virtuální měřicí/řídicí karty", které obsahují simulační model a nemuseli tento ovladač pro každý model psát v programovacím jazyce C "ručně", vyvinuli jsme speciální program, který nám **umožní vývoj tohoto ovladače automatizovat**. Máme tedy nyní možnost **bezpro-**

středně ze simulinkového schématu generovat zdrojový text příslušného virtuálního ovladače v C. Tím je možné jednoduše a rychle modifikovat ovladač pro prostředí Control Web při nejrůznějších úpravách a nových verzích simulačního modelu.

V prostředí Control Web jsme např. vytvořili simulátor Golem (Kofránek a spol 2001).

Další platformou pro vývoj simulátorů, kterou nyní v naší laboratoři využíváme převážně, je platforma **Microsoft .NET** a programovací prostředí Microsoft Visual Studio .NET, které, zejména ve své poslední verzi, poskytuje velké možnosti pro programátorskou práci.

V tomto prostředí již nejsme omezeni "předpřipravenými" prvky uživatelského rozhraní jako v prostředí Control Webu a přitom můžeme využívat veškerou sílu moderního nástroje pro tvorbu softwarových aplikací, na druhé straně si však musíme řadu prvků vytvářené aplikace naprogramovat sami.

Abychom si vytváření simulátorů ulehčili (a aby nebylo nutné ve Visual Studiu .NET "ručně" programovat již odladěný simulační model) vyvinuli jsme i zde speciální softwarový nástroj [17, 22], který automaticky ze Simulinku vygeneruje simulační model ve formě komponenty pro prostředí .NET (viz obr. 20).

Pohyblivé animace jako loutky na nitích simulačních modelů

Pro vytváření uživatelského rozhraní výukového simulátoru je velmi vhodné simulátor navenek reprezentovat jako pohyblivý obrázek. Proto simulační model propojíme s multimediální animací vytvořenou pomocí Adobe Flash [17].

Pro profesionální výsledný vzhled aplikace je nezbytné, aby vlastní animace vytvářel výtvarník – výsledky jsou neporovnatelně lepší, než když animace vytváří graficky nadaný programátor. Znamenalo to ovšem věnovat určité úsilí výuce výtvarníků, kteří musí zvládnout práci s nástroji pro tvorbu interaktivní grafiky. Takto vzdělaných výtvarníků je ale na trhu práce kritický nedostatek. Proto jsme již před lety začali úzce spolupracovat s výtvarnou školou Václava Hollara a na této škole jsme otevřeli "laboratoř interaktivní grafiky" jako naše detašované pracoviště. Iniciovali jsme také založení Vyšší odborné školy, která vyučuje v tříletém studiu předmět "interaktivní grafika". Pracovníci naší laboratoře se v této Vyšší odborné škole mimo jiné podílejí i na výuce. Animované obrázky mohou být řízeny výstupy implementovaného simulačního modelu a graficky reprezentovat význam číselných hodnot – např. schematický obrázek cévy se může roztažovat nebo komprimovat, plicní sklípek může hlouběji či mělčeji "dýchat", ručička měřicího přístroje se může pohybovat a průběžně zobrazovat hodnotu nějaké výstupní proměnné modelu čtené z běžícího simulačního modelu na pozadí.

Na druhé straně můžeme přes vizuální prvky vytvořené pomocí Adobe Flash (nejrůznější tlačítka, knoflíky, táhla apod.) do simulačního modelu zadávat nejrůznější vstupy.

Příklady simulátorů, vytvořených v prostředí Control Web a v prostředí Visual Studio .NET s propojením na flashové animace jsou uvedeny na obr. 21 a 22.

V případě složitější architektury může být logika propojení flashové animace a simulačního modelu poměrně složitá, proto je vhodnější mezi vrstvu vizu-

álních elementů a vrstvu simulačního modelu vložit řídicí vrstvu, která na jednom místě řeší veškerou logiku komunikace uživatelského rozhraní s modelem a kde je ukládán i příslušný kontext. V literatuře se hovoří o tzv. MVC architektuře výstavby simulátorů (Model – View – Controller).

Toto uspořádání je nezbytné zejména při složitějších modelech a simulátorech, jejichž uživatelské zobrazení je reprezentováno mnoha virtuálními přístroji na více propojených obrazovkách. Výhody tohoto uspořádání zvláště vyniknou při modifikacích jak modelu, tak i uživatelského rozhraní (obr. 23).

Při návrhu řídicí vrstvy, propojující vrstvu simulačního modelu s uživatelským rozhraním, se nám velmi osvědčilo využít propojené stavové automaty (jejichž pomocí je možno zapamatovat příslušný kontext modelu a kontext uživatelského rozhraní). Vytvořili jsme proto speciální softwarový nástroj, pomocí kterého můžeme propojené stavové automaty vizuálně navrhovat, interaktivně testovat jejich chování a automaticky generovat zdrojový kód programu pro prostředí Microsoft .NET [22]. Tento nástroj umožňuje zefektivnit programování propojek simulačního modelu s vizuálními objekty uživatelského rozhraní ve výukovém simulátoru.

Simulační hry na WEBU

Další technologický problém, který bylo při tvorbě výukových simulátorů nutno vyřešit, bylo proto nalezení způsobu, jak včlenit simulátory jako součásti internetových e-learningových aplikací. Při jeho řešení je možno postupovat několika způsoby. Jedním z nich je spuštění modelu na serveru a na klientský počítač posílat pouze výstupy z modelu. Znamená to ale pro každého připojeného uživatele na serveru individuálně spouštět jednu instanci (jeden exemplář) simulačního modelu. Kromě toho, zvláště u vzdálenějších a pomalejších připojení, je nezanedbatelné časové zpoždění mezi výstupem simulačního modelu na serveru a vykreslením výsledku na počítači klienta. Proto jsme se rozhodli jít spíše cestou spuštění simulačních modelů na počítači klienta.

U jednodušších modelů je možno využít interpretovaný ActionScript ve Flashi v němž je naprogramován simulační model. Složitější simulační modely už ale vyžadují kompilovaný simulátor.

Naše řešení [22] je zobrazeno na obr. 24. Uživatel si nainstaluje platformu .NET (pokud ji již nemá). Dále si na svém počítači nainstaluje speciální klientský program – "dispečer simulačních modelů". V případě potřeby se dispečer rozšíří i o runtime pro prostředí ControlWeb, které zajistí možnost spuštění modelů v tomto prostředí vytvořených. Webová e-learningová aplikace umožní stáhnout potřebné soubory simulačního modelu a dispečer simulačních modelů je spustí. Dispečer tak funguje obdobně jako prohlížeč obrázků, jen s tím rozdílem, že místo zobrazení obrázků spustí simulační program.

Zabalení simulačních her do multimediálního výkladu

Jedním z projektů, který vytváříme v naší laboratoři je již výše zmíněný Atlas fyziologie a patofyziologie (viz www.physiome.cz/atlas). Atlas [18] je průběžně vytvářená internetová multimediální výuková pomůcka z oblasti normální a patologické fyziologie, která s

využitím simulačních modelů pomáhá vysvětlit funkci a poruchy jednotlivých fyziologických systémů.

Simulační hry (obr. 25) jsou součástí e-learningových multimediálních výukových lekcí, jejichž podkladem je scénář vytvořený zkušeným pedagogem. Pedagog navrhuje vysvětlující text a s textem propojené doprovodné obrázky a animace. Animace jsou vytvářeny v prostředí Adobe Flash v úzké spolupráci pedagoga s výtvarníkem.

Text je poté namluven a synchronizován se spouštěním jednotlivých animací a s odkazy na simulační hry. Jednotlivé komponenty jsou kompletovány do výukových lekcí.

Pro vytváření a kompletaci multimediálních výukových lekcí využíváme softwarové prostředí serveru **Adobe Connect** (dříve Macromedia Breeze).

Projekt Atlasu fyziologie a patofyziologie je otevřený. Veškeré výukové texty, interaktivní animace a simulační modely včetně jejich zdrojových kódů jsou k dispozici všem zájemcům. Jako open-source jsou i veškeré námi vytvořené podpůrné softwarové nástroje, umožňující vygenerování jádra simulátorů z prostředí Matlab/Simulink a distribuci simulátorů prostřednictvím Internetu. Náš vývojový tým uvítá spolupráci s dalšími pracovišti na dalším rozšiřování atlasu a vytvoření jeho jazykových mutací.

Od entuziazmu k technologii a multidiscipinální spolupráci

Navzdory tomu, že se využití počítačů ve výuce stalo tématem řady konferencí, odborných i popularizačních článků, přesto, že hardwarové možnosti i softwarové nástroje dnes již dospěly do úrovně umožňující vytvářet náročná interaktivní multimedia, k výraznému rozšíření multimediálních výukových programů ve výuce medicíny zatím nedošlo.

Příčin je několik.

- Za prvé, ukazuje se, že tvorba výukových programů je podstatně náročnější na čas, lidské i materiální zdroje, než je obvykle plánováno
- Za druhé – tvorba kvalitních medicínských výukových programů vyžaduje týmovou multidisciplinární spolupráci zkušených pedagogů, lékařů, matematiků, fyziků, programátorů i výtvarníků.
- Konečně, pro kreativní propojení různých profesí, podílejících se na tvorbě výukové multimediální aplikace, musí být k dispozici vhodně zvolené výukové nástroje (jejichž ovládnutí ale vyžaduje určité úsilí a čas).
- Nároky stoupají, pokud na pozadí výukového programu má běžet simulační program, umožňující interaktivní simulační hry – ve vývojovém týmu pak musí být i odborníci, kteří jsou schopni navrhnout, formalizovat a odlatit příslušné modely (lékaři, matematici, fyzici a informatici).

Domníváme se, že nejdůležitějším výsledkem, kterého se nám v naší laboratoři zatím podařilo dosáhnout je vybudování multidisciplinárního týmu lékařů, matematiků, programátorů i výtvarníků, který je schopen výše zmíněné bariéry překonat.

Mezioborový tým potřebuje nástroje pro usnadnění komunikace a kooperace práce na společně vytvářených projektech. Pro podporu a koordinaci týmové spolupráce dnes existuje řada softwarových prostředků. V naší laboratoři se nám osvědčilo využití open-source nástroje WikiDoc, který umožňuje jednoduše využí-

vat webové rozhraní pro vzájemnou komunikaci členů týmu. Pro letný pohled do "kuchyně" našeho mezioborového týmu i pro seznámení s funkcí "wiki" – rozhraní se čtenář může podívat na "wiki-web" naší laboratoře: <http://physiome.ct/wiki>.

Nová výzva i příležitost pro vysoké školy

Současná epocha je charakterizována zásadními změnami v technologiích, které ve svém důsledku mění ekonomiku, společnost i způsob života. Původní soupeření o tuny produktů přešlo do soubroje o dokonalejší a rychlejší informace.

Vytváří se nový tržní segment, kde se obchoduje s nehmotným produktem, myšlenkami, nápady a know-how. Pokrok v technologiích vytváří tlak na flexibilitu pracovní síly a zvyšuje požadavek na průběžné rekvalifikace. Celoživotní vzdělávání se stává nutností ve stále větším počtu oborů.

Vytváření a realizaci rekvalifikačních kurzů a výuky v procesu celoživotního vzdělávání usnadňuje a podporuje využití informačních technologií. E-learning proto umožňuje rozšířit kapacitu vysokých škol a zároveň jim může přinést další příjmy z tvorby distančních rekvalifikačních a specializačních postgraduálních programů. Z tohoto hlediska je e-learning pro vysoké školy nová výzva, která bude vyžadovat velké úsilí, ale zároveň je i velkou příležitostí pro jejich další rozvoj.

Zdá se, že pomalu končí doba, kdy vytváření výukových programů bylo otázkou entuziazmu a pile skupin nadšenců. Tvorba moderních biomedicínských výukových aplikací je náročný a komplikovaný projekt, vyžadující **týmovou spolupráci** řady profesí – zkušených pedagogů vytvářejících základní scénář, tvůrců simulačních modelů, lékařů, výtvarníků a programátorů. Aby tato interdisciplinární kolektivní tvorba byla efektivní, je nutno pro každou etapu tvorby využívat **specifické výukové nástroje**, s dostatečnou technickou podporou, které umožňují interaktivní tvorbu simulačních modelů, vytváření interaktivních multimedií a jejich závěrečné propojení podle daného scénáře do kompaktního celku. K ovládnutí těchto nástrojů je zapotřebí věnovat značné úsilí, které se ale nakonec vyplatí.

Vytvořit kvalitní výukové e-learningové aplikace, pokrývající nezanedbatelnou část lékařského oboru není v silách jednoho pracoviště a MEFANET je zde velkou příležitostí.

Proto uvítáme spolupráci v rámci sítě MEFANET se všemi jednotlivci i pracovišti, kteří budou mít zájem. Rádi se podělíme o naše zkušenosti včetně námi vytvořených podpůrných softwarových prostředků.

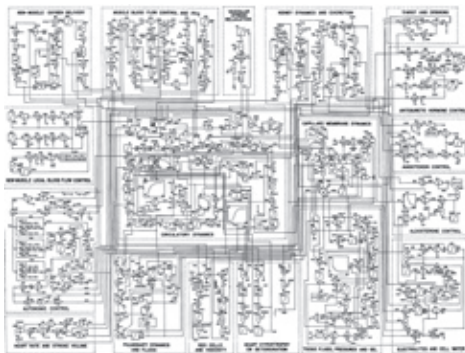
Literatura

- [1] Abram, S. R., Hodnett, B. L., Summers, R. L., Coleman, T. G., Hester R.L. (2007). Quantitative Circulatory Physiology: An Integrative Mathematical Model of Human Physiology for medical education. *Advanced Physiology Education*, 31 (2), 202–210.
- [2] Amosov, N. M., Palec B. L., Agapov, B. T., Jermakova, I. I., Ljabach E. G., Packina, S. A., Solovjev, V. P. (1977): Teoretičeskoje issledovanie fiziologičeskich sistēm. *Naukova Dumka, Kijev*, 1977.
- [3] Bassingthwaighte J. B. (2000). Strategies for the Physiome Project. *Annals of Biomedical Engineering* 28, 1043–1058.
- [4] Cameron, W.H. (1977): A model framework for computer simulation of overall renal function. *J. Theor. Biol. vol. 66.*, s. 552–572

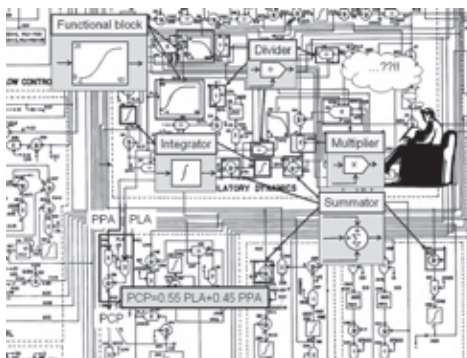
- [5] Coleman, T. G. and J.E. Randall (1983): HUMAN. A comprehensive physiological model. *The Physiologist*, vol. 26, (1): 15–21.
- [6] P. Fritzon (2003). Principles of Object-Oriented Modeling and Simulation with Modelica 2.1, Wiley-IEEE Press.
- [7] Grodins, F., S., J. Buell and A. Bart (1967): Mathematical analysis and digital simulation of the respiratory control system. *J. Appl. Physiol.*, vol. 22, (2): s. 250–276.
- [8] Guyton AC, Coleman TA, and Grander HJ. (1972): Circulation: Overall Regulation. *Ann. Rev. Physiol.*, 41, s. 13–41.
- [9] Guyton AC, Jones CE and Coleman TA. (1973): Circulatory Physiology: Cardiac Output and Its Regulation. Philadelphia: WB Saunders Company, 1973.
- [10] Guyton AC (1990): The suprising kidney-fluid mechanism for pressure control – its infinite gain!. *Hypertension*, 16, s.725–730.
- [11] Hunter P. J., Robins, P., & Noble D. (2002) The IUPS Physiome Project. *Pflugers Archive-European Journal of Physiology*, 445, s.1–9
- [12] Hunter, P. J. & Borg, T. K. (2003). Integration from proteins to organs: The Physiome Project. *Nature Reviews Molecular and Cell Biology*. 4, 237–243.
- [13] Ikeda N, Marumo F and Shirsataka M. A (1979): Model of Overall Regulation of Body Fluids. *Ann. Biomed. Eng.* 7, s. 135–166
- [14] Kofránek, J. Anh Vu, L. D., Snášelová, H., Kerekeš, R., & Velan, T (2001): GOLEM – Multimedia simulator for medical education. In Patel, L., Rogers, R., Haux R. (Eds.). *MEDINFO 2001, Proceedings of the 10th World Congress on Medical Informatics*. London: IOS Press, 1042–1046.
- [15] Kofránek, Jiří, Andrlík, Michal, Kripner, Tomáš, and Mašek, Jan (2002). From Simulation chips to biomedical simulator. In Amborski, K. and Meuth, H. *Proc. of 16th European Simulation Multiconference*, Darmstadt, 2002. SCS Publishing House, 2002, s. 431–436.
- [16] Kofránek, Jiří, Kripner, Tomáš, Andrlík, Michal, and Mašek, Jan. (2003) Creative connection between multimedia, simulation and software development tools in the design and development of biomedical educational simulators. *Simulation Interoperability Workshop*, 2003. Orlando, 2003, Position papers, Volume II, paper 03F-SIW-102, s. 677–687.
- [17] Kofránek J, Andrlík M, Kripner T, Stodulka P. (2005): From Art to Industry: Development of Biomedical Simulators. *The IPSI BgD Transactions on Advanced Research 2 (Special Issue on the Research with Elements of Multidisciplinary, Interdisciplinary, and Transdisciplinary: The Best Paper Selection for 2005)*, s.62–67.
- [18] Kofránek, J., Matoušek, S., Andrlík, M., Stodulka, P., Wunsch, Z., Privitzer, P., Hlaváček, J., Ondřej Vacek, O. (2007): Atlas of physiology – internet simulation playground. In. *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation*, Vol. 2. Full Papers (CD). (B. Zupanic, R. Karba, S. Blažič Eds.), University of Ljubljana, ISBN 978-3-901608-32-2, MO-2-P7-5, 1–9. 2007. Článek je dostupný na adrese http://patf-biokyb.lf1.cuni.cz/wiki/media/clanky/development_of_web_accessible_medical_simulators.pdf?id=wiki%3Auser%3Aseznam_publicaci&cache=cache
- [19] Jiří Kofránek, Jan Rusz, Stanislav Matoušek (2007): Guytons Diagram Brought to Life – from Graphic Chart to Simulation Model for Teaching Physiology. In *Technical Computing Prague 2007. 15th Annual Conference Proceedings*. Full paper CD-ROM proceedings. (P. Byron Ed.), Humusoft s.r.o. & Institute of Chemical Technology, Prague, ISBN 978-80-7%8%-658-6, 1–13, 2007. Článek, včetně zdrojových textů programů je dostupný na adrese <http://www.humusoft.cz/akce/matlab07/sbor07.htm#k>
- [20] Miller, J. A., Nair, R. S., Zhang, Z., Zhao, H. (1997). JSIM: A JAVA-Based Simulation and Animation Environment, In *Proceedings of the 30th Annual Simulation Symposium*, Atlanta, Georgia, 31–42.
- [21] Raymond, G. M., Butterworth E, Bassingthwaighte J. B. (2003). JSIM: Free Software Package for Teaching Physiological Modeling and Research. *Experimental Biology* 280,102–107
- [22] Stodulka, P., Privitzer, P., Kofránek, J., Tribula, M., Vacek, O. (2007): Development of WEB accessible medical educational simulators. In. *Proceedings of the 6th EUROSIM Congress on Modeling and Simulation*, Vol. 2. Full Papers (CD). (B. Zupanic, R. Karba, S. Blažič Eds.), University of Ljubljana, ISBN 978-3-901608-32-2, MO-3-P4-2, 1–6. 2007. Článek je dostupný na adrese http://patf-biokyb.lf1.cuni.cz/wiki/_media/clanky/development_of_web_accessible_medical_simulators.pdf?id=wiki%3Auser%3Aseznam_publicaci&cache=cache

Poděkování

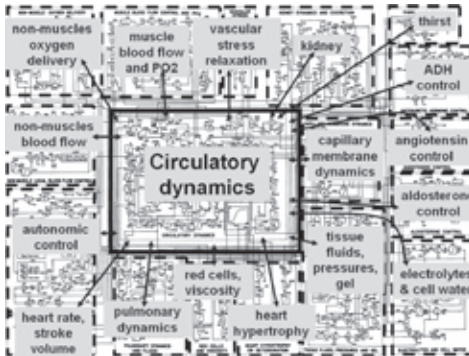
Práce na vývoji lékařských simulátorů je podporována grantem MŠMT č.2C06031 a společností BAJT servis s.r.o.



Obrázek 1: Grafické schéma rozsáhlého modelu cirkulace A. C. Guytona a spol. z roku 1972, publikovaného v *Annual Review of Physiology*. Předtištěno s laskavým svolením *Annual Review of Physiology*.



Obrázek 2: Detail Guytonova modelu z roku 1972. Model byl pro běžného čtenáře fyziologických časopisů na první pohled poněkud nepřehledný. Nebyl však jen směsicí čar a symbolů. Reprézentoval matematické vztahy, vyjádřené pomocí propojených symbolů pro funkční závislosti (Function block), dělení (Divider), násobení (Multiplier), integrování (Integrator), sčítání a odčítání (Summator). Tak například propojením tří bloků se dala graficky vyjádřit závislost pulmonárního kapilárního tlaku (PCP) na středním tlaku v levé síni (PLA) a na středním tlaku v pravé síni (PPA) vyjádřená vztahem $PCP = 0,55 PLA + 0,45 PPA$.



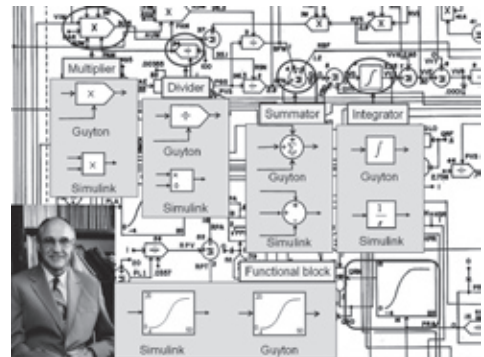
Obrázek 3: Guytonův model vyjadřoval zpětnovazebné propojení osmnácti subsystémů, ovlivňujících funkci cirkulačního systému, jako graficky zobrazenou soustavu matematických vztahů.



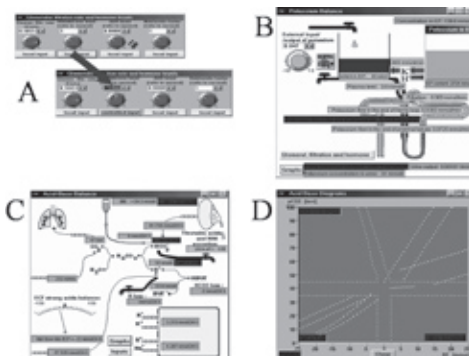
Obrázek 6: Dva typy problémů při tvorbě lékařských simulátorů a e-learningových programů využívajících simulační hry.



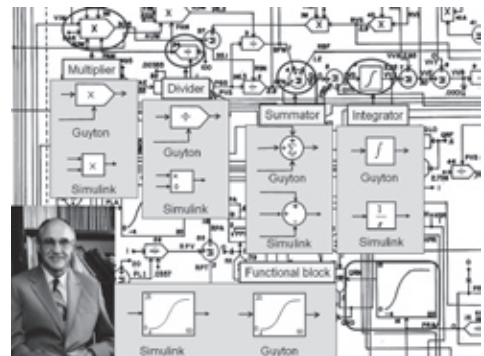
Obrázek 4: Spojení internetu, multimediálního prostředí, sloužícího jako zvukové a vizuální uživatelské rozhraní, se simulačními modely umožňuje studentům po připojení do kouzelné internetové pavučiny si názorně "osahat" vykládaný problém ve virtuální realitě. Staré Komenského krédo – "škola hrou" tak dnes nachází své moderní uplatnění.



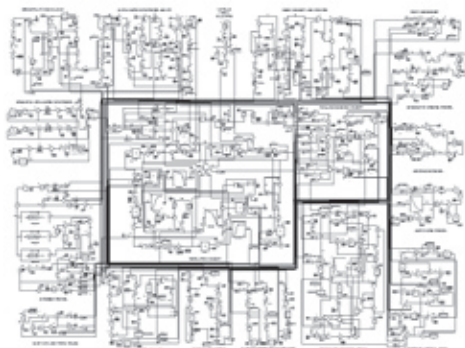
Obrázek 7: Grafické vyjádření násobiček (Multiplier), dělíček (Divider), sumátorů (Summator), integrotů (Integrator) a funkčních bloků (Functional block) v grafické notaci, navržené A.C.Guytonem (fotografie vlevo dole) a v nástroji pro tvorbu simulačních modelů Simulink od firmy Matworks.



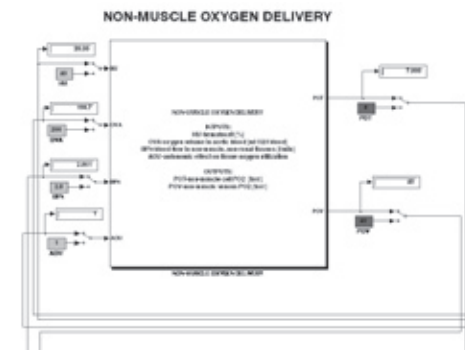
Obrázek 5: Příklad využití rozpojení fyziologických subsystémů z vnějších regulačních smyček v simulátoru GOLEM. Stisknutím tlačítka odpojíme řízení hladiny hormonu aldosteron z regulace. Pootočením knoflíku pak hladinu aldosteronu násilně zvýšíme (A). Po čase vidíme na schématu vylučování draslíku v ledvinách, že draslík se ve zvýšené míře vylučuje močí. Jeho zásohy v plazmě jsou však malé a ledvinné ztráty pak vedou k tomu, že draslík se přesouvá do mimobuněčné tekutiny (a plazmy) z buněk (B). Výstup draslíku z buněk je však provázen vstupem vodíkových iontů do buněk (C), důsledkem je rozvoj extracelulární metabolické acidózy (D).



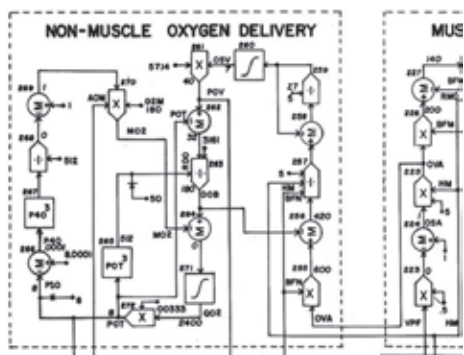
Obrázek 8: Oprava chyb v grafickém schématu modelu A.C.Guytona z roku 1972.



Obrázek 9: Implementace Guytonova modelu z roku 1972 v moderním nástroji pro tvorbu simulačních modelů Matlab/Simulink.



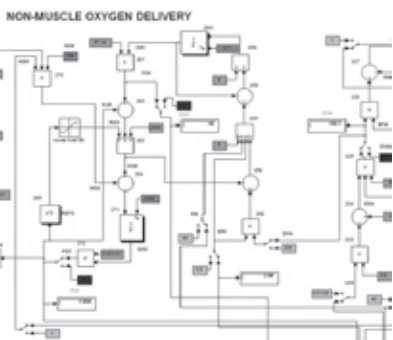
Obrázek 12: Subsystém dodávky kyslíku do „nesvalových tkání“ vyjádřený jako simulační čip. Zapojení odpovídá stejnému schématu jako na předchozím obrázku, vnitřní struktura počítačích prvků je však skryta uvnitř simulačního čipu. Model je pak „čitelnější“ a srozumitelnější i pro experimentální fyziology.



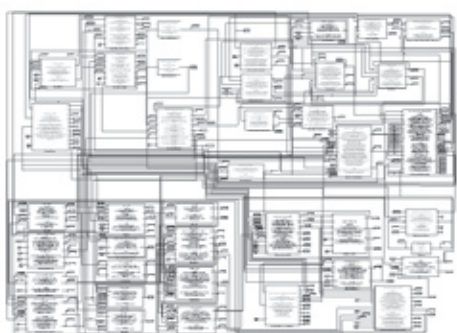
Obrázek 10: Pravá horní část grafického znázornění Guytonova modelu z roku 1972: Subsystém dodávky kyslíku do „nesvalových tkání“.



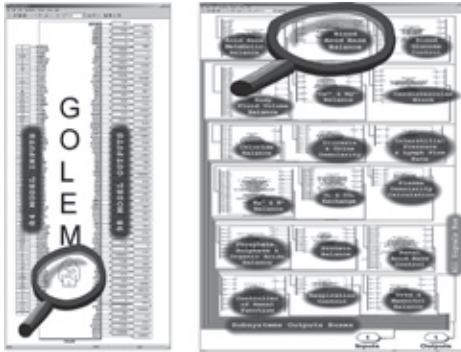
Obrázek 13: Implementace Guytonova modelu z roku 1972 v prostředí Matlab/Simulink, vyjádřená jako propojené simulační čipy. Každý ze simulačních čipů odpovídá jednomu z 18 propojených subsystémů.



Obrázek 11: Část modelu A.C. Guytona z roku 1972 v nástroji pro tvorbu simulačních modelů, která odpovídá grafickému znázornění z předchozího obrázku. Implementace subsystému dodávky kyslíku do „nesvalových“ tkání v prostředí Matlab/Simulink.



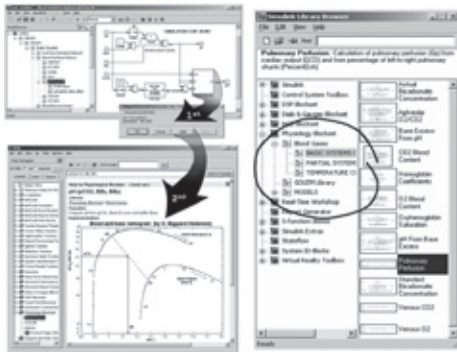
Obrázek 14: Implementace Guytonova modelu z roku 1986 v prostředí Matlab/Simulink ve formě simulačních čipů. Matematické vztahy jsou ukryty v simulačních čipech, propojení simulačních čipů znázorňuje uvažované vazby mezi jednotlivými subsystémy. Zdrojový text programu, včetně podrobného popisu významu všech rovnic je k dispozici na adrese <http://physiome.cz/guyton>.



Obrázek 15: Ukázka hierarchického uspořádání simulačního čipu, reprezentujícího simulační model, který je podkladem pro simulátor GOLEM. V prostředí Simulinku je možno snadno otestovat jeho chování – k jednotlivým "vstupním pinům" lze přivést vstupní hodnoty (nebo průběhy hodnot) a od "výstupních pinů" na virtuálních displejích či osciloskopech odečítat výstupy, resp. časové průběhy výstupů. Vpravo je zobrazen „vnitřek“ simulačního čipu nejvyšší hierarchické úrovně. Struktura připomíná elektrickou síť s propojenými integrovanými obvody, které v daném případě reprezentují simulační čipy nižší hierarchické úrovně. V následujícím obrázku je znázorněn obsah čipu "Blood Acid Base Balance".



Obrázek 16: Simulační čipy mají hierarchické uspořádání. Na obrázku je znázorněn "vnitřek" jednoho ze simulačních čipů z předchozího obrázku. Vzhledem k tomu, že každý jednotlivý simulační čip obsahuje dostatečně podrobnou dokumentaci o svých vstupech a výstupech, může být struktura vztahů uvnitř simulačního čipu (reprezentující fyziologické vztahy v reálném organismu) srozumitelná fyziologům. Na dalším obrázku je zobrazen obsah čipu "BEINV".

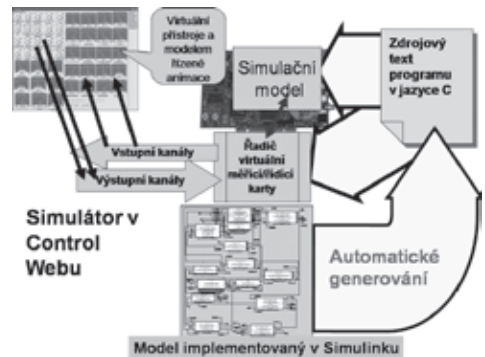


Obrázek 17: Simulační čipy na nejnižší hierarchické rovní jsou tvořeny propojenými základními komponenty vývojového systému Matlab/Simulink (sumátory, násobičkami, integrátory aj.) a případně i simulačními čipy dalších nižších úrovní. Tyto

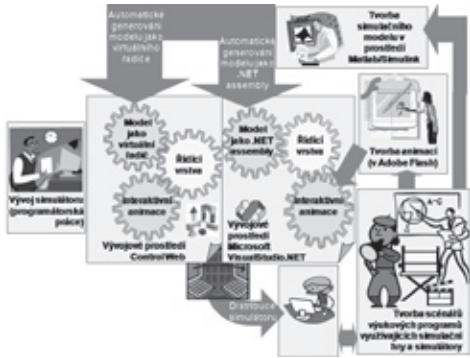
propojené prvky reprezentují jednotlivé matematické vztahy. Ke každému čipu je dynamicky připojena příslušná dokumentační stránka, která obsahuje věcný popis funkce čipu včetně popisu matematických vztahů, které jsou jejím podkladem. Simulační čipy mohou být ve vývojovém prostředí Simulink soustředovány do hierarchicky uspořádaných knihoven (viz obrázek vpravo). Z nich pak lze jednotlivé čipy "vytahovat" pomocí myši (jako z palety nástrojů), umísťovat je do vytvářené aplikace, propojovat je a vytvářet složitější modely. Implementace námi vytvořené knihovny simulačních čipů (obsahujících, mimo jiné i simulační čip „Golem“, je volně stažitelná z adresy <http://physiome.cz/simchips>.



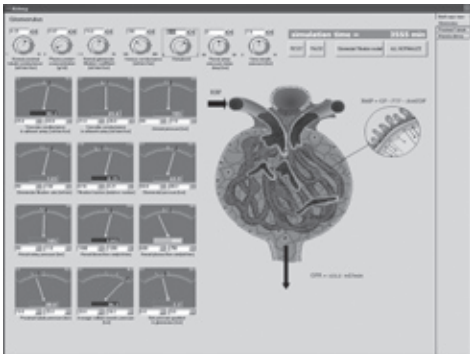
Obrázek 18: Komunikace systému Control Web s ovladačem řídicí měřicí karty při tvorbě průmyslových aplikací. Měřicí ústředny či velín průmyslové aplikace, vytvořené v prostředí Control Web komunikuje přes vstupní a výstupní kanály s řídicím měřicí/řídicí kartou, která komunikuje s propojeným technologickým zařízením.



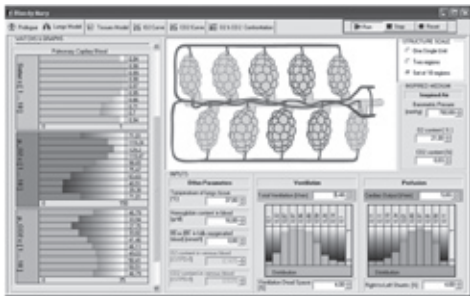
Obrázek 19: Začlenění simulačního modelu do ovladače "virtuální karty" při tvorbě simulátoru v prostředí Control Web. Simulátor vytvořený v prostředí Control Web komunikuje přes vstupní a výstupní kanály s řídicím virtuální (v hardwarové podobě neexistující) měřicí/řídicí kartou. V tomto řídicí je „ukryt“ simulační model. Vstupní proměnné jsou simulačnímu modelu „posílány“ přes výstupní kanály aplikace v Control Webu. Výstupy modelu jsou čteny, obdobně jako měřené výstupy technologického zařízení, prostřednictvím vstupních kanálů. Aby nebylo nutno ručně programovat řídicí s „ukrytým“ simulačním modelem, vyvinuli jsme speciální nástroj, který automaticky generuje zdrojový text řídicí v jazyce C přímo z modelu, implementovaného v Simulinku.



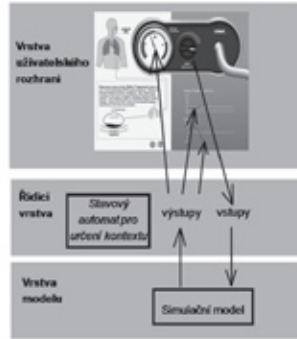
Obrázek 20: Kreativní propojení nástrojů a aplikací pro tvorbu simulátorů a výukových programů využívajících simulační hry. Základem e-learningového programu kvalitní scénář, vytvořený zkušeným pedagogem. Tvorba animovaných obrázků je odpovědnost výtvarníků, kteří vytvářejí interaktivní animace prostředí Adobe Flash. Jádrem simulátorů je simulační model, vytvářený v prostředí speciálních vývojových nástrojů, určených pro tvorbu simulačních modelů. V naší laboratorii zde využíváme prostředí Matlab/Simulink od firmy Matworks (a v poslední době i nástroje pro práci se simulačním jazykem Modelica). Vývoj simulátoru je náročná programátorská práce, pro jejíž usnadnění jsme vyvinuli speciální programy, usnadňující automatický převod vytvořeného simulačního modelu z prostředí Matlab/Simulink do prostředí Control Web a Microsoft .NET.



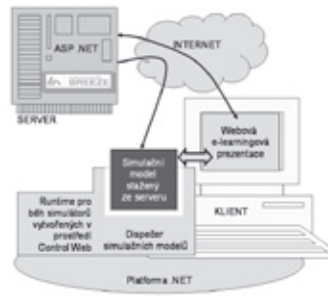
Obrázek 21: Příklad simulátoru vytvořeného v prostředí ControlWeb. Výstupy modelu jsou zobrazovány na ručkových měřicích přístrojích a zároveň ovlivňují i tvar animovaného obrázku ledvinného glomerulu (velikost, tloušťku šipek a číselnou hodnotu aj.), vytvořeného pomocí programu Adobe Flash.



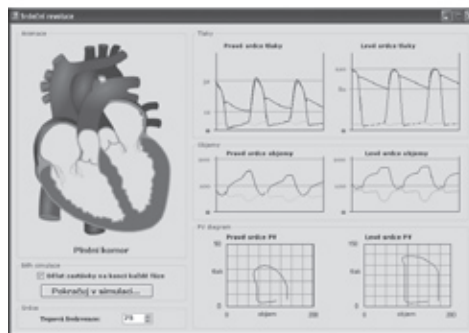
Obrázek 22: Příklad simulátoru vytvořeného v prostředí Microsoft Visual Studio .NET. Výstupy modelu přenosu krevních plynů jsou zobrazovány v číselné podobě i pomocí sloupcových diagramů. Zároveň model ovlivňuje i tvar animovaného obrázku schematicky znázorňujícího distribuci ventilace a perfúze v plicích, který byl vytvořen pomocí programu Adobe Flash.



Obrázek 23: Tzv. MVC architektura při tvorbě simulátorů. Mezi vrstvou modelu a vrstvou uživatelského rozhraní je vhodné vložit řídicí vrstvu, kam jsou směřovány veškeré zprávy a události vznikající ve virtuálních přístrojích uživatelského rozhraní, a kam je zároveň směřována veškerá komunikace s modelem. V této vrstvě se řeší veškerý kontext zobrazovaných dat a příslušné požadavky na komunikaci s modelem. Veškerá logika zobrazování a komunikace je pak soustředěna do jednoho místa, což podstatně ušetří čas při modifikacích uživatelského rozhraní nebo změnách modelu.



Obrázek 24: Využití simulačních modelů v internetových e-learningových kurzech. Dispečer simulačních modelů funguje obdobně jako prohlížeč obrázků, jen s tím rozdílem, že místo zobrazení obrázků spustí simulační program.



Obrázek 25: Ilustrativní ukázka doprovodného simulátoru ve výukovém programu patofyziologie cirkulace.